대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0021502
Application Number

출 원 년 월 일 : 2003년 04월 04일
Date of Application APR 04, 2003

출 원 인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2004 년 03 월 25 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】 명세서 등 보정서

【수신처】 특허청장

【제출일자】 2003.05.28

【제출인】

【명칭】 삼성전자 주식회사

【출원인코드】 1-1998-104271-3

【사건과의 관계】 출원인

【대리인】

【성명】 김동진

【대리인코드】 9-1999-000041-4

【포괄위임등록번호】 2002-007585-8

【사건의 표시】

【출원번호】 10-2003-0021502

【출원일자】 2003.04.04

【발명의 명칭】 플래시 메모리의 오류블록 관리방법

【제출원인】

【접수번호】 1-1-2003-0121066-58

【접수일자】 2003.04.04

【보정할 서류】 명세서등

【보정할 사항】

【보정대상항목】 별지와 같음

【보정방법】 별지와 같음

【보정내용】 별지와 같음

【취지】 특허법시행규칙 제13조·실용신안법시행규칙 제8조의 규정에의하여 위와 같 이 제출합니다. 대리인
김동진 (인)

【수수료】

【보정료】 0 원

【추가심사청구료】 0 원

【기타 수수료】 0 원

【합계】 0 원

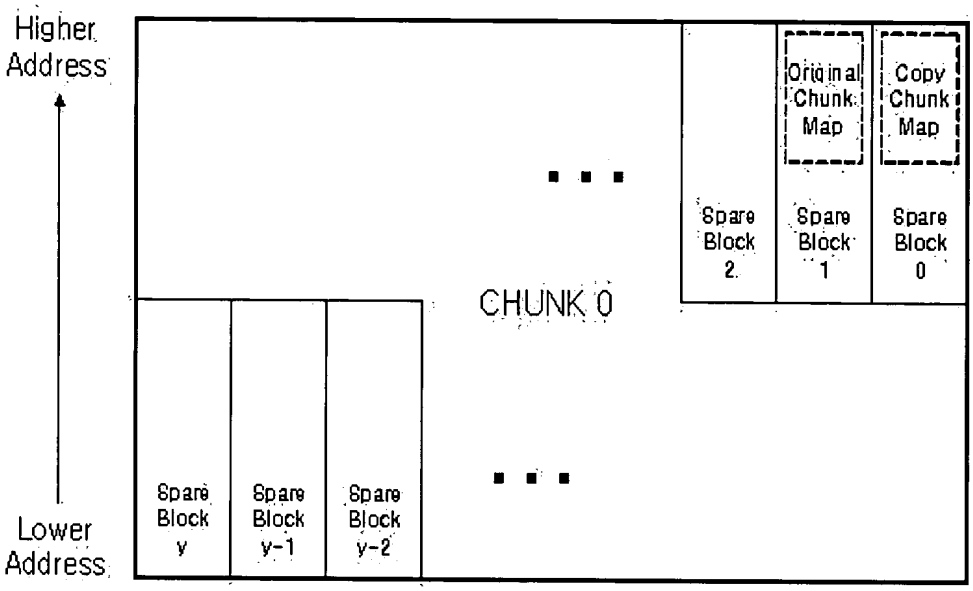
【첨부서류】 1. 보정내용을 증명하는 서류_1통

【보정대상항목】 도 1

【보정방법】 정정

【보정내용】

【도 1】



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.04.04
【발명의 명칭】	플래시 메모리의 오류블록 관리방법
【발명의 영문명칭】	METHOD FOR MANAGING BAD BLOCK IN FLASH MEMORY
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	김동진
【대리인코드】	9-1999-000041-4
【포괄위임등록번호】	2002-007585-8
【발명자】	
【성명의 국문표기】	윤송호
【성명의 영문표기】	YOON,Song Ho
【주민등록번호】	741030-1245819
【우편번호】	420-031
【주소】	경기도 부천시 원미구 상1동 반달마을 극동신라아파트 1843-707
【국적】	KR
【발명자】	
【성명의 국문표기】	김장환
【성명의 영문표기】	KIM,Jang Hwan
【주민등록번호】	730810-1074418
【우편번호】	138-908
【주소】	서울특별시 송파구 잠실1동 주공아파트 34동 208호
【국적】	KR
【발명자】	
【성명의 국문표기】	김범수
【성명의 영문표기】	KIM,Bum Soo
【주민등록번호】	690121-1019710

【우편번호】 431-080
【주소】 경기도 안양시 동안구 호계동 1055-1 무궁화아파트 703동 203호
【국적】 KR
【발명자】
【성명의 국문표기】 인지현
【성명의 영문표기】 IN,Ji Hyun
【주민등록번호】 771124-2063411
【우편번호】 120-090
【주소】 서울특별시 서대문구 홍제동 무악청구아파트 110동 802호
【국적】 KR
【발명자】
【성명의 국문표기】 정태선
【성명의 영문표기】 CHUNG,Tae Sun
【주민등록번호】 710304-1030617
【우편번호】 156-034
【주소】 서울특별시 동작구 상도4동 279-400
【국적】 KR
【취지】 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 김동진 (인)
【수수료】
【기본출원료】 20 면 29,000 원
【가산출원료】 16 면 16,000 원
【우선권주장료】 0 건 0 원
【심사청구료】 0 항 0 원
【합계】 45,000 원
【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】

【요약】

플래시 메모리의 사용 중에 발생하는 오류블록을 관리하는 방법을 제공한다. 플래시 메모리 오류블록 관리 방법은 플래시 메모리를 사용블록을 포함하는 사용영역과 오류블록의 발생을 대비한 예비블록들을 포함하는 예비영역으로 할당하는 단계와, 할당된 예비영역에 오류블록 및 이를 대체하는 예비블록과의 매핑정보를 포함하는 블록맵페이지그룹을 생성하는 단계, 및 사용 중 생성된 오류블록을 예비블록으로 대체시키고 블록맵페이지그룹을 갱신하는 단계를 포함한다.

본 발명에 따라 최소의 플래시 연산으로 오류블록을 관리할 수 있다.

【대표도】

도 12

【색인어】

플래시 메모리, Flash Memory, Bad Block, 오류블록, 매핑

【명세서】

【발명의 명칭】

플래시 메모리의 오류블록 관리방법{METHOD FOR MANAGING BAD BLOCK IN FLASH MEMORY}

【도면의 간단한 설명】

도 1은 종전 발명의 플래시 메모리 시스템에서 오류블록 관리의 기본 단위인 청크(Chunk)의 구조를 보여주는 블록도이다.

도 2는 도 1에 도시된 청크맵을 보다 상세한 구조를 보여주는 블록도이다.

도 3은 본 발명의 시스템 구성요소를 보여주는 블록도이다.

도 4는 k개의 플래시 메모리 칩으로 구현한 플래시 메모리의 블록도이다.

도 5는 t개의 사용영역의 파티션과 한 개의 예비영역의 파티션으로 나누어진 플래시 메모리 칩을 보여주는 블록도이다.

도 6은 비보호영역과 보호영역을 모두 포함하는 플래시 메모리의 어느 한 칩의 구조를 보여주는 블록도이다.

도 7은 모든 영역이 비보호영역으로 이루어진 플래시 메모리의 어느 한 칩의 구조를 보여주는 블록도이다.

도 8은 y개의 블록으로 나누어진 한 파티션의 구조를 보여주는 블록도이다.

도 9는 비보호영역과 보호영역을 모두 포함하는 예비영역의 구조를 보여주는 블록도이다.

도 10은 전 영역이 비보호영역으로 이루어진 예비영역의 구조를 보여주는 블록도이다.

도 11은 도 9와 도 10의 블록맵페이지그룹의 구조를 상세히 보여주는 블록도이다.

도 12는 플래시 메모리의 어느 한 칩에 대한 플래시 장치관리자의 동작 프로세스를 보여주는 흐름도이다.

도 13은 도 12의 예비영역포맷 프로세스를 보다 상세히 보여주는 흐름도이다.

도 14는 도 12의 블록맵정보의 램상주 프로세스를 보다 상세히 보여주는 흐름도이다.

도 15는 도 12의 오류블록처리 프로세스를 보다 상세히 보여주는 흐름도이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<16> 본 발명은 플래시 메모리 사용 중 발생되는 오류블록을 처리하는 방법에 관한 것으로 보다 상세하게는 플래시 연산을 최소화하는 플래시 메모리 오류블록 관리 방법에 관한 것이다.

<17> 플래시 메모리는 RAM이나 ROM 또는 하드디스크와 같이 데이터 저장수단이나 플래시 메모리는 복수의 블록을 갖고있으며 블록단위로 지우기 연산을 한다는 점에서 특징을 갖는다.

<18> 플래시 메모리의 각 블록은 지우기 연산 횟수의 제한이 있다. 그래서 플래시 메모리가 공장에서 출하된 이후에 지우기 연산의 횟수가 제한을 넘게 되면, 해당 블록은 오류블록이 될 수 있으며, 또한 물리적 특성이 좋지 않는 블록인 경우, 지우기 연산 횟수가 제한 횟수에 접근하게 되면 해당 블록은 오류블록(Bad Block)이 될 수 있다. 그리고, NAND 플래시 메모리 같은 플래시 메모리는 공장에서 출하될 때부터 몇몇 블록은 오류블록일 수 있다. 또한, 플래시 메모리에 읽기 연산을 할 때 이전에 플래시 메모리에 쓰였던 데이터와 다른 데이터가 읽힐 수도 있으며, 해당 블록 또한 오류블록이다. 이러한 오류블록들은 플래시 메모리를 사용 중에 모든 블

록에서 발생할 수 있으며, 또는 사용하기 전부터 오류블록일 수 있다. 오류블록의 존재는 플래시 메모리 사용을 어렵게 만드는 요소이다.

<19> 이러한 오류블록을 관리하기 위해서는 오류블록을 관리하기 위한 정보가 필요하며 이런 정보를 RAM에 또는 EEPROM과 같은 별도의 메모리에 저장하여 관리할 수 있다. 그러나 RAM에 관리 정보를 저장했을 경우에는 전원이 꺼지면 RAM에 저장된 정보는 모두 잃어버리게 되는 문제점이 있고, EEPROM과 같은 별도의 메모리에 관리 정보를 저장하는 경우는 시스템 구축을 위한 추가 비용이 필요하고 시스템을 복잡하게 만들 수 있다.

<20> 이러한 문제점을 해결하기 위하여 미국특허 6,260,151 B1에 개시된 종전 발명은 플래시 메모리 시스템에서 오류블록(Bad Block)을 처리하는 방법을 제시하였는데 플래시 메모리의 일정한 영역을 오류블록 관리를 위한 부분으로 할당하여 Bad Block을 처리한다.

<21> 도1은 상기 발명의 오류블록 관리의 기본 단위인 청크(Chunk)의 구조를 보여주는 블록도이다. 종전 발명은 플래시 메모리 영역을 실제 데이터를 쓰고 지우는데 사용되는 데이터 영역과 오류블록이 발생된 경우에 이를 처리하기 위한 여분의 공간인 여분영역(Spare Area)으로 구분하고, 여분영역은 여러 개의 청크로 구분하였다. 도1은 청크의 구조를 보여주고 있다. 청크는 여분블록 0부터 여분블록 y까지의 y+1개의 여분블록으로 이루어져 있다. 청크 하나에는 메모리 번지수가 큰 영역에 오리지날청크맵(Original Chunk Map) 및 카피청크맵(Copy Chunk Map)이 각각 하나의 여분블록에 저장된다. 그리고 비휘발성 메모리인 램에는 워킹청크맵(Working Chunk Map)이 존재한다.

<22> 도2는 청크맵의 보다 상세한 구조를 보여준다. 즉, 여분블록 0을 확대한 그림이다. 청크맵은 플래시 메모리에 존재하는 오리지날청크맵, 카피청크맵, 및 램에 존재하는 워킹청크맵 모두가 같은 데이터 구조를 갖는다. 우선 오류블록관리 헤더(BBM Header) 부분에는 오류블록을 관리

하기 위한 메타 정보가 저장되는데 예를 들면, 총 블록의 수, 청크맵의 크기, 청크맵의 상태 등이 있다. 그리고 각각 유효필드(Valid Field)와 오류데이터블록번호(Bad Data Block Number)를 갖는 엔트리(BBM Entry)가 여분블록의 개수와 같은 y+1개가 있다. 엔트리는 오류블록과 이를 대체하는 여분블록의 매팽 정보가 저장된다. 예를 들어 7번 엔트리의 오류데이터블록번호가 10이라면 오류블록인 10번 블록이 청크 n의 7번 여분블록에 재매핑(Re-mapping)되어 있음을 나타낸다.

<23> 상기 종래기술의 동작은 다음과 같다. 먼저 오류블록 관리자를 초기화하는데, 오류블록 관리자의 초기화 과정은 오류블록 관리자포맷, 청크포맷, 청크마운트 과정으로 이루어지는데 오류블록 관리자의 초기화가 완료되는 시점은 청크가 마운트되는 시점이다. 청크를 마운트하는 과정은 플레시 메모리의 카피청크맵과 오리지날청크맵을 읽어서 램상에 워킹청크맵을 생성하는 과정이다. 이 때 플래시 메모리에 카피청크맵과 오리지날청크맵 두개를 유지하여 각종 연산 중에 오류가 발생하더라도 복구할 수 있는 방법을 제공한다. 오류블록 관리자 초기화 작업이 끝나면 준비 상태(Ready State)가 된다. 특정 블록에 대한 접근 방법은 각 청크에 대하여 요청된 블록이 엔트리에 존재하는지 검사한 후 만일 엔트리에 존재하면 재매핑한 결과를 보내준다. 새로운 오류블록이 발견되었을 때의 알고리즘을 요약하면 다음과 같다. 우선 사용할 수 있고 비사용 엔트리를 찾고 그 엔트리를 포함하는 청크맵을 읽어서 워킹청크맵에 쓴다. 다음으로 새로운 매핑정보를 워킹청크맵의 찾아진 비사용 엔트리에 쓴다. 그리고 나서 변경된 재매핑 정보를 플래시 메모리의 오리지날청크맵 및 카피청크맵에 기록한다.

<24> 상술한 종래 기술은 플래시 메모리상의 카피청크맵과 오리지날청크맵 및 램상의 워킹청크맵이 일관성을 유지하도록 여러 번의 플래시 메모리에 대한 쓰기와 지우기 연산을 수행한다. 예를 들어 새로운 실행시간 오류블록이 발생했을 경우 다음과 같은 연산을 필요로 한다.

<25> 1. 오리지날청크맵 블록 지우기 연산

<26> 2. 오리지날청크맵 블록에 오리지날청크맵 쓰기 연산

<27> 3. 오리지날청크맵의 상태를 무효(Invalid)로 쓰기 연산

<28> 4. 카피청크맵 블록 지우기 연산

<29> 5. 카피청크맵 블록에 카피청크맵 쓰기 연산

<30> 6. 오리지날청크맵의 상태를 유효(Valid)로 쓰기 연산

<31> 즉, 플래시 메모리에 두 번의 지우기 연산과 네 번의 쓰기 연산을 하게 된다. 또한 오류블록과 여분블록의 매핑 정보가 여러 개의 청크에 저장되기 때문에 오류블록을 처리하기 위해서 각 청크를 마운트하고 각 청크에 존재하는 청크맵간의 일관성을 유지하기 위하여 복잡한 알고리즘을 수행하여야 한다. 따라서 실행시간 오류블록이 발생하여도 플래시 연산의 횟수를 많이 필요로 하지 않는 발명이 필요하다.

<32> 최근에는 플래시 메모리를 단순히 데이터를 저장하기 위한 공간으로 사용하는 것을 넘어서 OS(Operating System) ROM 이미지나 부트로더(Boot Loader)처럼 갱신이 필요없는 데이터를 저장하는 경우가 생겼다. 갱신할 필요가 없는 데이터가 적힌 플래시 메모리의 영역은 쓰기나 지우기 연산을 할 필요가 없을 뿐만 아니라, 다른 데이터가 해당 영역에 기록되거나 플래시 메모리 관련 소프트웨어의 잘못된 동작이나 사용자의 실수에 의해 발생될 수 있는 지우기 연산을 방지할 필요성이 있다. 이러한 필요에 따라 읽기 전용 영역을 설정하는 기능이 플래시메모리 혹은 플래시메모리 콘트롤러에 제공되는 경우에 읽기 전용 영역에서 생긴 오류블록을 대체하는 새로운 읽기 전용 영역을 설정할 수 있는 방법 및 시스템이 필요하다.

【발명이 이루고자 하는 기술적 과제】

<33> 본 발명의 목적은 오류영역이 발생했을 때 적은 플래시 연산의 횟수로 오류영역을 관리할 수 있는 방법 및 시스템을 제공하는 것이다.

<34> 본 발명의 다른 목적은 플래시 메모리의 일정 영역이 읽기 전용으로 보호되고 있는 경우에 이 영역에서 발생하는 오류블록에 대한 관리 방법 및 시스템을 제공하는 것이다.

【발명의 구성 및 작용】

<35> 상기 목적을 달성하기 위하여 본 발명인 플래시 메모리의 오류블록 관리방법은 플래시 메모리에 복수의 사용블록을 갖는 사용영역과 복수의 예비블록을 갖는 예비영역을 할당하는 하는 단계, 상기 할당된 사용영역 또는 예비영역에서 발생하는 오류블록과 상기 발생된 오류블록을 대체하여 사용하는 예비블록과의 매핑정보를 저장하기 위한 블록맵페이지그룹을 상기 예비영역에 생성하는 단계, 및 오류블록이 발생할 때마다 상기 매핑정보를 이용하여 사용가능한 예비블록을 찾아 상기 오류블록 대신에 상기 예비블록을 사용할 수 있도록 상기 블록맵페이지그룹에 저장된 매핑정보를 갱신하는 단계를 포함한다. 한편, 플래시 메모리가 복수의 플래시 칩으로 이루어진 경우에 각각의 칩에 대해서 상기 단계를 수행한다.

<36> 블록맵페이지 그룹을 생성하는 단계는 할당된 예비영역의 특성에 따라 블록맵정보를 생성하고 블록맵정보에 저장된 정보를 초기화하는 단계, 상기 할당된 예비영역의 오류블록을 검색하여 오류블록과 이를 대체하는 예비블록의 번호를 기록하여 블록맵정보를 생성하는 단계, 및 및 생성된 블록맵정보를 상기 블록맵페이지그룹에 기록하는 단계를 포함한다. 또한, 사용영역 및 상기 예비영역은 양자 모두 읽기전용의 보호영역과 읽기, 쓰기 및 지우기가 가능한 비보호영역을 갖도록 할당할 수도 있다.

<37> 블록맵페이지그룹은 예비블록의 매핑정보들을 모두 포함하는 블록맵페이지를 적어도 하나 이상 포함하는 동일한 포맷의 블록을 적어도 하나 이상 포함하도록 생성하며, 바람직하게는 2개의 블록으로 생성한다. 상기 블록맵페이지는 유효여부를 표현하는 트랜지션필드, 다른 블록맵페이지와의 저장된 정보의 선후를 알려주는 카운트필드, 및 예비블록의 숫자만큼의 블록맵필드를 포함하도록 생성한다. 각 블록맵필드는 오류블록번호와 이를 대체하는 예비블록번호를 기재하는 오류블록번호필드와 예비블록번호필드를 포함한다. 블록맵페이지그룹은 읽기오류가 발생할 때 오류블록을 예비블록으로 대체하지 않고 오류블록을 복구할 수 있도록 하기 위하여 트랜스퍼블록을 더 포함하도록 생성할 수 있다.

<38> 상기 블록맵페이지그룹을 갱신하는 단계는 사용가능한 예비블록을 찾는 단계, 오류블록과 예비블록의 매핑정보를 갖는 블록맵정보를 생성하는 단계, 상기 생성된 블록맵정보를 바탕으로 상기 블록맵페이지그룹을 갱신하는 단계, 상기 오류블록의 데이터를 사용가능한 예비블록에 복사하는 단계를 포함한다. 상기 블록맵페이지를 갱신할 때는 상기 블록맵페이지그룹에서 사용가능한 블록맵페이지를 찾는 단계와, 상기 사용가능한 블록맵페이지에 상기 블록맵정보를 쓰는 단계, 및 상기 블록맵페이지에 유효마크하는 단계를 거친다.

<39> 이하, 첨부도면을 참조하여 본 발명에 따른 바람직한 실시예를 상세히 설명한다.

<40> 도 3의 시스템은 플래시 메모리(100), 플래시 메모리 콘트롤러(200), 중앙처리장치(300), 및 XIP(Execute-In-Place)가능한 메모리(400)를 포함한다. XIP(Execute-In-Place)가능한 메모리(400)란 롬이나 램처럼 그 안에서 소프트웨어를 실행시킬 수 있는 메모리(이하, "주메모리"라고 함)를 말한다.

<41> 중앙처리장치(300)는 주메모리(400)에 상주한 플래시 어플리케이션 및 플래시 장치관리자를 이용하여 플래시 메모리(100)에 필요한 정보를 쓰고 읽거나 삭제할 수 있으며(이하, "플래시 연산"이라고 함), 삭제할 때는 블록단위로 수행해야 한다.

<42> 플래시 메모리 콘트롤러(200)는 플래시 메모리의 일정 영역을 읽기 전용 영역으로 만들거나 플래시 연산성능을 향상시키는 등의 여러 가지 기능을 제공하나 플래시 메모리 콘트롤러(200)를 제외한 중앙처리장치(300)와 주메모리(400) 및 플래시 메모리(100)만으로도 시스템을 구현할 수 있다. 한편, NOR형 플래시 메모리를 사용할 경우에는 NOR형 플래시 메모리는 XIP가 가능한 메모리이므로 이 때는 주메모리(400)를 빼고 플래시 메모리(100)와 중앙처리장치(300)만으로도 시스템을 구현할 수 있다.

<43> 주메모리(400)에는 플래시 장치관리자나 플래시 어플리케이션 및 플래시 메모리의 사용 중에 발생되는 오류블록을 관리할 수 있는 오류블록관리 프로그램이 상주할 수 있다. 플래시 장치관리자는 오류블록의 관리방법을 소프트웨어로 구현한 모듈로서, 주메모리(400)에 상주하며 중앙처리장치(300)가 플래시 장치관리자의 코드를 실행시킴으로써 오류블록을 관리한다. 플래시 어플리케이션은 플래시 장치관리자를 사용하는 상위계층이며 부트로더(Boot-loader), 플래시 트랜슬레이션 레이어(Flash Translation Layer), 파일 시스템 등이 될 수 있다. 플래시 장치관리자는 플래시 어플리케이션이 플래시 메모리(100)를 사용할 수 있게 해준다. 오류블록관리 프로그램은 플래시 어플리케이션이 플래시 메모리(100)를 사용하다가 발생된 오류블록을 관리하는 모든 프로세스를 정의하는 프로그램이다.

<44> 도 4는 k개의 플래시 메모리 칩으로 구현한 플래시 메모리의 블록도이다. 플래시 메모리(100)는 k개의 플래시 메모리 칩으로 구현할 수 있는데 각 메모리 칩은 사용영역(110)과 오류블록을 관리하기 위한 예비영역(120)을 가질 수 있다. 물론 본 발명에서는 오류블록을 관리하기 위한

예비영역을 특정 칩, 예를 들어 칩1, 칩2, 칩3에 몰아 넣을 수도 있다. 각 칩이 예비영역(120)을 갖도록 함으로써 플래시 연산을 위해 항상 오류블록유무를 검사해야 하는 범위를 하나의 칩에 한정함으로써 플래시 연산의 성능을 높일 수 있다. 예비영역(120)의 크기는 각 칩에서 발생할 수 있는 최대 오류블록수와 오류블록을 관리하기 위해 필요한 블록수의 합으로 결정된다. 예비영역(120)에서 필요한 블록의 수를 결정하는 방법은 후술한다.

<45> 도 5는 t개의 사용영역의 파티션과 한 개의 예비영역의 파티션으로 나누어진 플래시 메모리 칩을 보여주는 블록도이다. 플래시 메모리에 부트로더롬이미지(Boot-Loader ROM Image), OS롬이미지(Operating System ROM Image), 또는 파일시스템 데이터 등을 저장하는데 있어 각각의 용도를 위해서 플래시 메모리에 영역을 구분하여 할당할 수 있고, 이렇게 구분된 영역을 파티션이라고 한다. 각 파티션은 파티션관리소프트웨어가 있으며 각 파티션의 소프트웨어는 자기의 파티션에서 발생할 수 있는 오류블록을 관리한다. 사용영역(110)은 해당 칩의 최하위주소부터 할당되고, 예비영역(120)은 사용영역(110)이 사용하지 않는 곳에 할당되나, 이 순서는 바꿀 수도 있다. 이 때 사용영역(110)은 t개의 파티션으로 나눌 수 있는데, 파티션의 개수 t는 플래시 메모리에 저장되는 정보의 종류에 따라 결정된다.

<46> 도 6은 비보호영역과 보호영역을 모두 포함하는 플래시 메모리의 어느 한 칩의 구조를 보여주는 블록도이다. 플래시 메모리(100) 또는 플래시 메모리 콘트롤러(200)가 플래시 메모리의 일정 영역을 읽기전용 영역(보호영역)으로 설정할 수 있다면 도6과 같은 구조로 해당 칩의 사용영역(110)과 예비영역(120)은 보호영역(130)과 비보호영역(140)으로 나눌 수 있다. 보호영역(130)에 대해서는 읽기 연산만 가능하며 쓰기 연산과 지우기 연산은 허용되지 않는다. 즉, OS롬이미지나 부트로더처럼 특별히 보호되어야할 데이터가 기록되는 영역이라고 할 수 있다. 비보호영역(140)은 읽기, 쓰기, 및 지우기가 모두 가능한 영역을 말한다. 도 6에서는 비보호영

역(140)는 사용영역(110)과 예비영역(120)에 걸쳐 있는 형태로 구현하였다. 보호사용영역은 최하위주소부터 할당되고 그 위에 비보호사용영역, 비보호예비영역, 및 보호예비영역이 차례로 할당된다. 그러나 이 순서를 바꾼 실시예도 가능하다.

<47> 도 7은 모든 영역이 비보호영역으로 이루어진 플래시 메모리의 어느 한 칩의 구조를 보여주는 블록도이다. 만일 OS롬이미지나 부트로더처럼 특별히 보호되어야할 데이터가 없는 경우라면 모든 영역을 비보호영역(140)으로 설정하여 사용할 수도 있다. 이런 경우에 모든 파티션은 플래시 연산이 가능하다.

<48> 도 8은 y개의 블록으로 나누어진 한 파티션의 구조를 보여주는 블록도이다. 블록은 플래시 메모리에서 지우기 연산의 단위로서 어떤 블록에 하나의 비트라도 오류가 생기면 해당 블록은 오류블록이 된다. 오류블록은 사용영역(110)뿐만 아니라 예비영역(120)에서도 발생할 수 있다.

<49> 도 9는 비보호영역과 보호영역을 모두 포함하는 예비영역의 구조를 보여주는 블록도이다. 예비영역(120)은 비보호영역(140)의 오류블록을 대체하기 위한 m개의 예비블록(154)와 보호영역(130)의 오류블록을 대체하기 위한 n개의 예비블록(155)을 포함하여 총 m+n개의 예비블록과, 사용영역(110) 또는 예비영역(120)의 오류블록과 이를 대체하기 위한 비보호 또는 보호예비블록(154 또는 155)을 매핑하는 정보를 담고 있는 비보호 또는 보호블록맵페이지그룹(150 또는 158)과, 읽기영역의 오류를 처리하기 위한 트랜스퍼블록(152), 및 파티션정보를 담고 있는 파티션정보블록(156)을 포함할 수 있다. 이 때 상기 n,과 m은 고정된 숫자가 아니다. 예를 들면, 총 m+n=100 일 때, 보호블록에서 오류가 발생하지 않는 동안에 100번 연속해서 비보호 블록에서 오류가 발생한 경우에 m=100, n=0이 될 수 있다.

<50> 블록맵페이지그룹은 사용영역(110) 및 예비영역(120) 안의 비보호영역(140)의 오류블록과 이를 대체하는 예비블록(154)과의 매핑정보를 담고 있는 비보호블록맵페이지그룹(150)과, 사용영역

(110) 및 예비영역(120) 안의 보호영역(130)의 오류블록과 이를 대체하는 예비블록(155)과의 매핑정보를 담고 있는 보호블록맵페이지그룹(158)을 포함하는 개념으로 사용한다. 각 블록맵페이지는 두 개의 블록으로 구성되는 것이 바람직하며 이에 대해서는 후술한다. 한편, 비보호블록맵페이지(150)와 보호블록맵페이지(158)은 하나의 블록맵페이지로 통합하여 사용할 수도 있는데, 이 때에도 두 개의 블록으로 구성되는 것이 바람직하다.

<51> 파티션정보블록(156)은 칩의 파티션에 관한 정보를 담고 있는 블록으로서 파티션에 대한 정보를 플래시 메모리에 저장하지 않는다면, 파티션정보블록(156)은 예비영역(120)에 할당되지 않을 것이다.

<52> 트랜스퍼블록(152)는 읽기 오류가 발생한 블록을 회복하기 위해서 사용하는 블록이다. 플래시 메모리에서 읽기 오류가 발생한 경우에 지우기 연산을 하고 나면 다시 사용할 수 있는 경우가 있으므로 읽기 오류블록을 재사용하기 위해서 트랜스퍼블록을 사용할 수 있다. 만일 읽기오류를 쓰기 오류 및 지우기 오류와 동일하게 취급하여 오류블록을 예비블록으로 대체할 경우 트랜스퍼블록(152)은 불필요하다.

<53> 도 10은 전 영역이 비보호영역으로 이루어진 예비영역의 구조를 보여주는 블록도이다. 이 때, n은 0이 되고 m은 플래시 메모리를 사용하는 동안에 발생해도 플래시 메모리의 성능이 보장되는 최대 오류블록의 수와 같다.

<54> 도 11은 도 9와 도 10의 블록맵페이지그룹의 구조를 상세히 보여주는 블록도이다. 블록맵페이지그룹은 q개의 블록맵페이지(160)로 구성되어 있고, 각 블록맵페이지(160)는 오류블록번호(166)와 이를 대체하는 예비블록번호(167)로 구성되는 블록맵필드(161)를 예비블록(154 또는 155)의 개수와 같은 m+n개를 갖는다. 또한 블록맵페이지(160)는 가장 최근에 쓰여진 블록맵페이지를 찾는데 쓰이는 카운트필드(163)와, 자신이 갖고 있는 블록맵필드정보(161)가 유효한

지, 무효인지, 아니면 사용되지 않고 있는지를 구별할 수 있도록 하는 트랜지션필드(165)도 포함한다. 가장 최근에 씌여진 블록맵페이지(160)는 트랜지션필드(165)에 유효표시를 갖는 블록맵페이지(160) 중에서 카운트필드(163)에 가장 큰 숫자가 씌여진 것으로서, 이것을 주메모리, 예를 들어 램에 올린 것을 블록맵정보(170)라고 한다. 블록맵정보(170)를 통한 블록맵페이지 그룹의 갱신은 후술한다. 한편, 본 발명의 바람직한 실시예에 있어서 오류블록번호(166)과 예비블록번호(167)은 모두 2바이트를 갖는데, 이는 현재 상용화된 플래시 메모리의 지우기 연산 가능횟수는 대략 1만에서 10만정도이다. 따라서 오류블록은 그 숫자보다는 상당히 적을 것이므로 2바이트는 0에서 65535까지의 숫자를 가질 수 있으므로 그 정도면 충분한 양이기 때문이다.

<55> 도 3의 주메모리(400)에 있는 플래시 어플리케이션은 사용영역(110)의 특정한 사용블록을 접근하기 위해 장치관리자에게 해당 사용블록의 논리주소를 보내고 장치관리자는 논리주소에 맞는 물리주소를 갖는 블록을 찾는다. 이 때 오류블록을 사용하지 않도록 하기 위해서 사용블록번호가 가장 최근의 오류블록정보를 담고 있는 블록페이지(160)의 블록맵정보(170)에 등록된 블록맵필드(161) 중 오류블록넘버(166)와 일치하면 이를 대체하는 예비블록(167)을 사용한다.

<56> 도 12는 플래시 메모리의 어느 한 칩에 대한 플래시 장치관리자의 동작 프로세스를 보여주는 흐름도이다. 도 13와, 도 14, 및 도 15는 각각 예비영역포맷 프로세스를 보다 상세히 보여주는 흐름도, 블록맵정보의 램상주 프로세스를 보다 상세히 보여주는 흐름도, 및 오류블록처리 프로세스를 보다 상세히 보여주는 흐름도이다. 본 발명의 바람직한 실시예는 각 칩마다 오류블록을 대비한 예비영역을 갖고있으므로 이하에서는 하나의 칩을 중심으로 설명한다. 어느 한 칩과 마찬가지로 다른 칩에서도 동일한 프로세스로 오류블록을 관리할 수 있다.

<57> 맨 처음 사용영역과 예비영역을 나누고 예비영역에 블록맵페이지그룹을 만드는 프로세스, 즉 예비영역을 포맷하는데(S100), 본 발명의 동작에 있어 처음 1회 때만 필요한 프로세스이다. 예비영역포맷(S100)의 동작은 도 13을 통해 설명한다.

<58> 예비영역을 포맷하기 위해서는 먼저 칩의 일정부분은 사용영역으로 할당하고 또 일정영역은 플래시 메모리 사용 중에 발생하는 오류블록에 대한 대비책으로 예비영역으로 할당한다(S102). 예비영역을 위하여 할당해야 할 블록의 개수는 플래시 메모리 사용 중에 발생할 수 있는 최대 오류블록의 개수와 발생되는 오류블록을 관리하기 위한 블록의 개수의 합으로 구할 수 있다. 전자에 할당된 블록은 예비블록들의 모임이고 후자에 할당된 블록은 블록맵페이지그룹을 포함한다. 오류블록을 관리하기 위한 블록의 숫자는 다음과 같이 할당할 수 있다. 먼저 칩을 보호영역과 비보호영역으로 나누어 사용할 경우에 보호블록맵페이지그룹과 비보호블록맵페이지그룹을 위하여 각각 2개의 블록을 할당하며, 칩을 비보호영역만으로 사용할 경우에는 비보호블록맵페이지그룹을 위한 2개의 블록만을 할당한다. 각 블록맵페이지그룹에 2개의 블록을 할당하는 이유는 후술한다.

<59> 파틴션 정보를 칩에 저장하고자 하는 경우에는 파티션정보블록을 한 블록 할당한다. 또한 읽기 오류가 발생한 블록을 예비블록으로 대체하지 않고 재사용할 수 있도록 하는데 필요한 트랜스퍼 블록도 한 블록을 할당한다. 이를 표로 정리하면 다음과 같다.

<60> 【표 1】

오류블록관리를 위하여 할당될 총 블록 수

비보호/보호영역지원	비보호영역지원	트랜스퍼 블록지원	파티션 정보블록지원	할당될 총 블록수
○		○	○	6
○		○		5
○			○	5
○				4
	○	○	○	4
	○	○		3
	○		○	3
	○			2

<61> 예비영역할당(S102)이 끝나면, 블록맵정보를 초기화한다(S104). 블록맵정보는 평상시에는 도 1의 주메모리(400)에 상주하는 오류블록관리 프로그램에 포함된 정보로서 원칙적으로 블록맵페이지그룹에 저장되어 있는 가장 최근의 블록맵페이지와 동일한 정보를 갖는다. 맨 처음 플래시 메모리를 사용할 때는 블록맵정보의 생성을 위하여 블록맵정보를 초기화해야 한다. 블록맵정보의 초기화는 블록맵정보의 크기를 결정하고 각 필드를 특정 값으로 초기화하는 것을 의미한다. 블록맵정보의 크기는 트랜지션필드(165)크기 + 카운트필드(163)크기 + 블록맵필드 한개의 크기*예비블록의 개수로 정해진다.

<62> 블록맵정보가 초기화되면(S104) 이를 바탕으로 블록맵정보를 생성한다(S106). 블록맵정보를 생성하려면 사용영역(110)과 예비영역(120)의 최하위주소부터 최상위주소까지 오류블록이 존재하는 지를 검사한다. 오류블록이 발견되면 블록맵정보에 포함된 블록맵필드에 발견된 오류블록과 이를 대체하는 예비블록의 번호를 적는다. 한편, 플래시 메모리를 보호영역과 비보호영역으로 나누어 사용하는 경우라면 각각의 블록맵정보(보호블록맵정보와 비보호블록맵정보)를

생성해야 한다. 물론 보호영역과 비보호영역으로 나누더라도 블록맵페이지그룹을 하나로 통합해서 사용하는 경우에는 블록맵정보를 나누어서 생성할 필요가 없다.

<63> 마지막으로 생성된 블록맵정보를 예비영역의 블록맵페이지그룹에 기록한다(S108). 기록하는 방법은 블록맵페이지그룹에 속한 모든 블록에 지우기 연산을 수행하고 가장 하위주소의 블록맵페이지에 블록맵정보를 기록하면 된다.

<64> 예비영역포맷이 완료되면(S100), 블록맵정보를 주메모리에 상주시킨다(S200). 처음 플래시 메모리를 사용하는 경우가 아니라면 블록맵정보는 가장 최근에 기록된 블록맵페이지와 같은 정보를 갖도록 최신의 블록맵페이지를 검색하여 상주시킨다(S200).

<65> 블록맵정보를 주메모리에 상주시키는 방법은 도 14를 참조하여 설명한다. 먼저 블록맵정보의 카운트필드에 0을 기록하여 카운트 필드를 초기화한다(S202). 그리고 첫 번째 블록맵페이지를 읽는다(S204). 읽은 블록맵페이지의 유효성을 검사하여(S206) 유효한 블록맵페이지라면 블록맵정보의 카운트와 읽은 블록맵페이지의 카운트를 비교한다(S214). 블록맵정보의 카운트가 작다면 읽은 블록맵페이지의 정보가 블록맵정보보다는 더 최근의 정보라는 것을 의미하므로 블록맵페이지를 블록맵정보에 복사하여 더 최근의 정보로 갱신한다(S216). 만일 읽은 블록맵페이지가 마지막 블록맵페이지라면(S218) 블록맵정보에 저장된 정보가 가장 최신의 블록맵페이지를 담고 있으므로 주메모리에 상주시킨다. 만일 마지막 블록맵페이지가 아니라면 더 최신의 블록맵페이지가 존재하는지를 찾기 위하여 다음 블록맵페이지를 읽

는다(S210). 그리고 나서 상기 설명한 과정을 반복한다. 유효 블록맵페이지인지를 검사하는 단계(S206)에서 유효페이지가 아니라면 이를 무시하고 블록웹페이지를 읽고(S210) 상기 설명한 과정을 반복한다. 만일 읽기오류를 쓰기나 지우기 오류와 별도로 취급한다면, 읽기 오류를 처리하기 위한 프로세스(S208, S212)가 추가된다. 읽기오류에 대한 처리과정은 다음과 같다. (1)먼저 트랜스퍼블록에 지우기 연산을 하고, (2)읽기오류 블록의 데이터를 정정한 후에 트랜스퍼블록으로 복사한다. (3)이 때 예비영역에서 사용할 수 있는 블록맵페이지를 찾아서 트랜지션필드에 읽기오류 복구상태에 있음을 표시하고, (4)읽기오류블록에 대한 지우기 연산을 한다. (5)그리고 나서 트랜스퍼블록의 데이터를 읽기오류블록으로 복사하고, (6)마지막으로 블록맵페이지에 무효마크를 한다. 읽기오류를 표시하는 (3)단계는 (4)단계이후에 복구작업 중 갑자기 전원이 차단되는 경우를 대비한 것이다. 마지막 무효마크를 하는 것은 이미 읽기오류블록은 이후에는 정상적으로 사용할 수 있기 때문에 읽기오류블록이 체크된 블록맵페이지를 무효마크하는 것이다. 유효블록맵페이지검사에서(S206) 읽기 오류가 표시된 블록맵페이지라면 (S208) 상기 (4), (5), (6)단계를 거쳐 오류블록을 복구하고(S212) 다음 블록맵페이지를 읽고 (S210) 다시 상기 S206에서 S218의 과정을 반복한다.

<66> 블록맵정보가 주메모리에 상주하면(S200) 플래시 연산을 하기 전에는 오류블록관리를 할 수 있는 상태가 된다(S300). 도 1의 주메모리(400)의 플래시 어플리케이션은 사용영역의 특정 주소를 플래시 연산을 하기 위해서는 플래시 장치관리자에 논리주소를 넘겨주고 플래시 장치관리자는 해당 논리주소를 바탕으로 물리적

주소를 생성한다(S400). 물리적 주소를 찾으면(S400) 플래시 연산을 한다(S500). 플래시 연산 중에 오류블록이 발견되면 해당 논리 주소를 오류블록이 아닌 예비블록에 매핑해주는 작업을 한다(S600). 오류블록을 처리하는 과정은 도 15를 참조하면 다음과 같다. 먼저 미사용 예비블록을 찾는다(S602). 사용가능한 예비블록은 블록맵정보를 이용하여 찾을 수 있다. 예비블록을 찾으면(S602), 블록맵정보를 갱신한다(S604). 블록맵정보의 갱신은 블록맵정보에 있는 찾아진 예비블록을 매핑하도록 예비블록과 오류블록의 번호를 적는다. 그리고 나서 블록맵페이지그룹에서 사용할 수 있는 블록맵페이지를 찾고(S606), 오류블록의 데이터를 미사용 예비블록으로 복사한다(S608). 그리고 나서 사용할 수 있는 블록맵페이지에 블록맵정보를 쓰고 (S610), 사용할 수 있는 블록맵페이지에 블록맵정보의 유효마크를 쓴다(S612). 오류블록을 예비블록으로 대체한 플래시 메모리는 오류블록이 존재하지 않는 것처럼 사용할 수 있다. 상기 S606단계에 쓰이는 블록맵페이지를 찾는 방법은 블록맵페이지의 트랜지션 필드를 사용하여 유효여부를 판단하고 유효한 것 중에서 카운트필드의 숫자가 가장 큰 것으로 판단한다. S610단계에서 빈 블록맵페이지에 블록맵정보를 기재하는데 이 때 기존에 저장된 블록맵페이지 중에서 가장 큰 카운트를 갖는 것보다 1만큼 큰 카운트를 카운트필드에 적는다. 만일 비어있는 블록맵페이지가 없다면, 다른 블록맵페이지를 찾아서 쓰는데 거기도 꽉 찬 상태라면 최신의 블록맵페이지가 있는 블록이 아닌 블록에 대하여 지우기 연산을 하고 기록하면 된다.

<67> 예를 들어, 10개의 블록맵페이지를 갖고 있는 A블록과 B블록이 있다고 하자. 블록맵페이지를 사용하는 순서는 최하위주소부터 최상위주소로 사용한다고 할

때, A블록의 블록맵페이지를 모두 사용한 경우에는 B블록의 블록맵페이지를 사용한다. 이 때 각각의 블록맵페이지는 생성 당시의 매핑정보를 가지고 있으므로 A블록과 B블록은 일종의 매핑정보 히스토리 블록역할을 한다. 따라서, A블록을 모두 사용하고 B블록의 마지막 블록맵페이지까지 사용한 경우에 새로운 오류블록이 발생하면 A블록을 지우고 거기에 새로운 매핑정보를 포함하는 블록맵페이지를 만든다. 이 때 만일 시스템의 오작동이나 전원이 차단되더라도 최소한 현재의 매핑정보를 제외한 가장 최근의 매핑정복 B블록에 존재하므로 플래시 메모리를 별 문제 없이 사용할 수 있다. 차후 현재 오류블록에 대해 플래시 연산을 하려고 하면 그 때 오류블록에 대하여 새롭게 예비블록을 매핑하고 그 매핑정보를 포함하여 블록맵페이지를 만들면 된다.

<68> 한편, 블록맵페이지그룹이 하나의 블록으로 되어 있다면 비어있는 블록맵페이지가 없을 때 블록맵페이지그룹을 지우고 다시 처음부터 쓸 수 있으나 만일 블록맵페이지를 지울 때 시스템의 오동작이나 전원이 꺼질 경우 기존의 매핑정보는 사라진다. 따라서 두 개 이상의 블록을 갖는 것이 바람직하다.

<69> 본 발명이 속하는 기술분야의 당업자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 예를 들어, 이상의 실시예에서는 각각의 칩이 모두 예비영역을 갖는 것을 중심으로 설명하였으나 본 발명은 이에 한정되지 않고 예비영역을 특정한 칩에 몰아서 사용하는 경우에도 적용될 수 있다.

<70> 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

【발명의 효과】

<71> 상기한 바와 같이 이루어진 본 발명에 따르면, 플래시 메모리의 사용 중에 오류블록이 생겼을 경우에 적은 플래시 메모리 연산으로 오류블록을 관리할 수 있어 시스템의 성능을 종전의 발명과 달리 향상 시킬 수 있다. 또한 오류블록 문제 해결에 대한 간단한 알고리즘을 제공함으로써 메모리 제약이 있는 내장형 플래시 메모리 시스템에서 유용하게 쓰일 수 있다.

<72> 또한 읽기전용으로 설정된 영역(보호영역)이 있는 플래시 메모리의 보호영역에서 발생된 오류블록을 대체하여 할당되는 새로운 영역 또한 읽기전용으로 설정 할 수 있어 변경할 필요가 없는 중요한 데이터를 안전하게 관리할 수 있다.

【특허청구범위】

【청구항 1】

(a) 플래시 메모리에 복수의 사용블록을 갖는 사용영역과 복수의 예비블록을 갖는 예비영역을 할당하는 하는 단계;

(b) 상기 할당된 사용영역 또는 예비영역에서 발생하는 오류블록과 상기 발생된 오류블록을 대체하여 사용하는 예비블록과의 매핑정보를 저장하기 위한 블록맵페이지그룹을 상기 예비영역에 생성하는 단계; 및

(c) 오류블록이 발생할 때마다 상기 매핑정보를 이용하여 사용가능한 예비블록을 찾아 상기 오류블록 대신에 상기 예비블록을 사용할 수 있도록 상기 블록맵페이지그룹에 저장된 매핑정보를 갱신하는 단계를 포함하는 플래시 메모리의 오류블록 관리방법

【청구항 2】

제1항에 있어서, 플래시 메모리가 복수의 플래시 칩으로 이루어진 경우에 상기 (a), (b), 및 (c) 단계는 각각의 칩에 대하여 이루어지는 것을 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 3】

제1항에 있어서, 상기 (b)단계는 할당된 예비영역의 특성에 따라 블록맵정보를 생성하고 블록맵정보에 저장된 정보를 초기화하는 (b1)단계;

상기 할당된 예비영역의 오류블록을 검색하여 오류블록과 이를 대체하는 예비블록의 번호를 기록하여 블록맵정보를 생성하는 (b2)단계; 및

상기 생성된 블록맵정보를 상기 블록맵페이지그룹에 기록하는 (b3)단계를 포함하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 4】

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 사용영역 및 상기 예비영역 양자 모두 읽기전용의 보호영역과 읽기, 쓰기 및 지우기가 가능한 비보호영역을 갖도록 할당하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 5】

제1항에 있어서, 상기 블록맵페이지그룹은 예비블록의 매핑정보들을 모두 포함하는 블록맵페이지를 적어도 하나 이상 포함하는 동일한 포맷의 2개의 블록으로 생성하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 6】

제5항에 있어서, 상기 블록맵페이지는 유효여부를 표현하는 트랜지션필드, 다른 블록맵페이지와의 저장된 정보의 선후를 알려주는 카운트필드, 및 예비블록의 숫자만큼의 블록맵필드를 포함하도록 생성하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 7】

제5항에 있어서, 상기 블록맵페이지그룹은 읽기오류가 발생할 때 오류블록을 예비블록으로 대체하지 않고 오류블록을 복구할 수 있도록 하기 위하여 트랜스퍼블록을 더 포함하도록 생성하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 8】

제6항에 있어서, 상기 각 블록맵필드는 오류블록번호와 이를 대체하는 예비블록번호를 기재하는 오류블록번호필드와 예비블록번호필드를 포함하도록 생성하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 9】

제1항에 있어서, 상기 (c) 단계는 사용가능한 예비블록을 찾는 (c1) 단계;

상기 오류블록과 상기 예비블록의 매핑정보를 갖는 블록맵정보를 생성하는 (c2) 단계;

상기 생성된 블록맵정보를 바탕으로 상기 블록맵페이지그룹을 갱신하는 (c3) 단계; 및

상기 오류블록의 데이터를 사용가능한 예비블록에 복사하는 (c4) 단계를 포함하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【청구항 10】

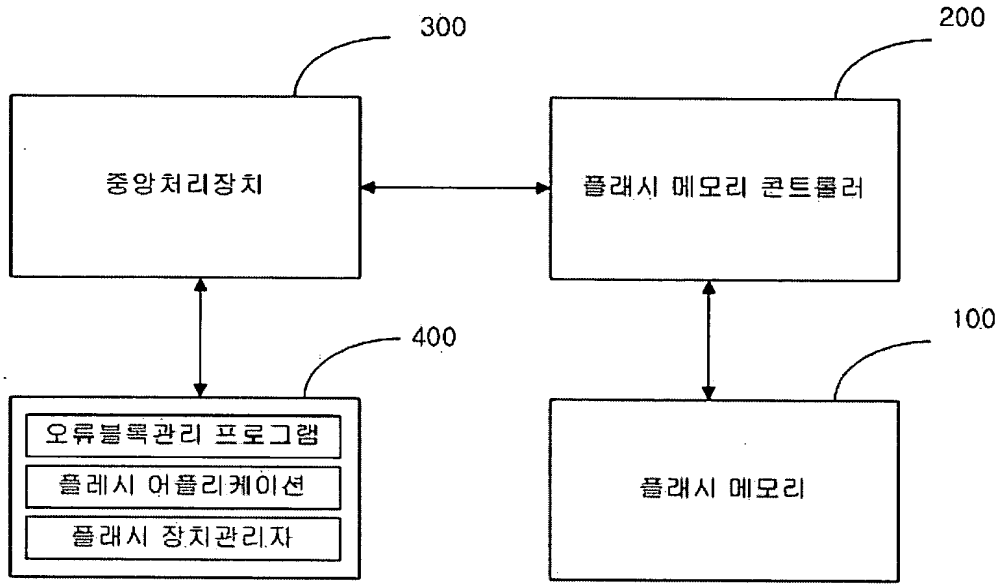
제9항에 있어서, 상기 (c3) 단계는 상기 블록맵페이지그룹에서 사용가능한 블록맵페이지를 찾는 (c31) 단계;

상기 사용가능한 블록맵페이지에 상기 블록맵정보를 쓰는 (c32) 단계; 및

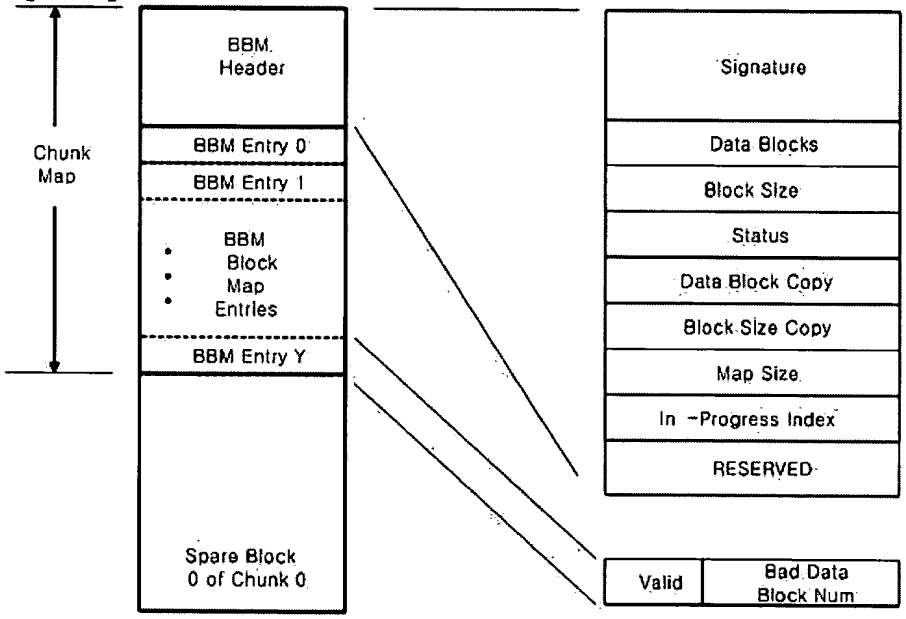
상기 블록맵페이지에 유효마크하는 (c33) 단계를 포함하는 것을 특징으로 하는 플래시 메모리의 오류블록 관리방법

【도면】

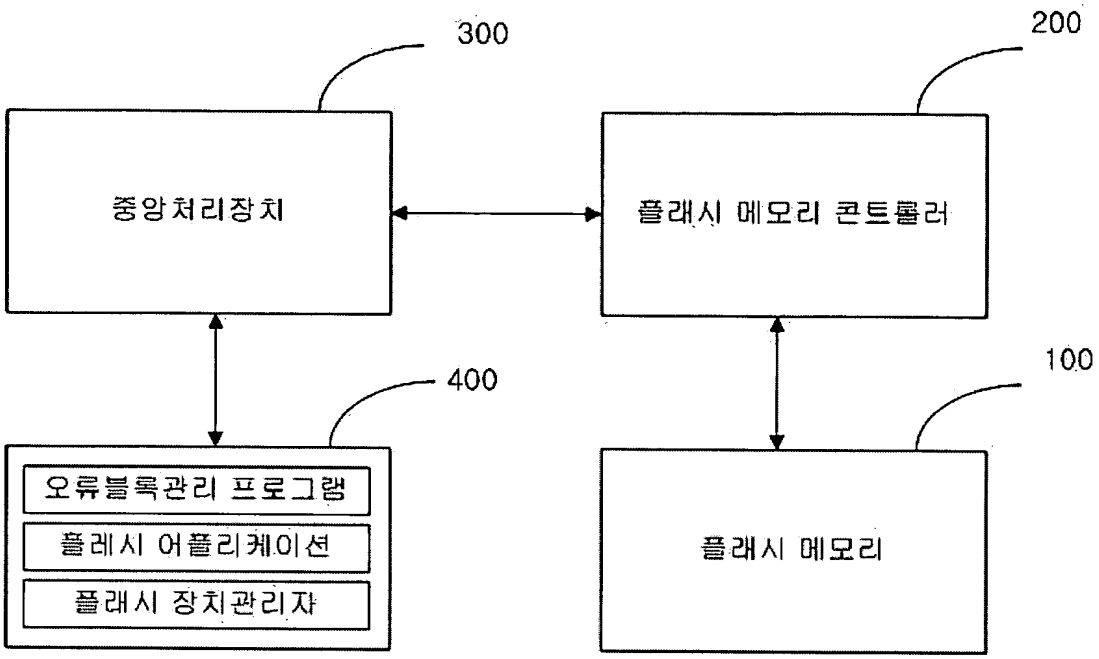
【도 1】


300
200
중앙처리장치
플래시 메모리 콘트롤러
400
100
오류블록관리 프로그램
플레시 어플리케이션
플래시 장치관리자
플래시 메모리

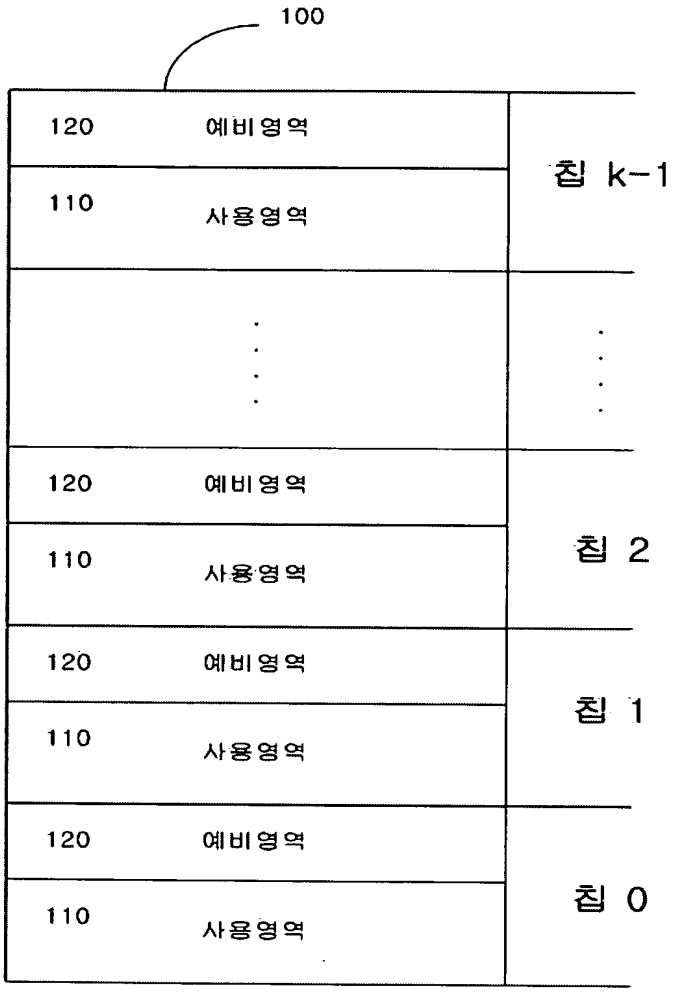
【도 2】


Chunk Map
BBM Header
BBM Entry 0
BBM Entry 1
BBM Block Map Entries
BBM Entry Y
Spare Block 0 of Chunk 0
Signature
Data Blocks
Block Size
Status
Data Block Copy
Block Size Copy
Map Size
In -Progress Index
RESERVED
Valid
Bad Data Block Num

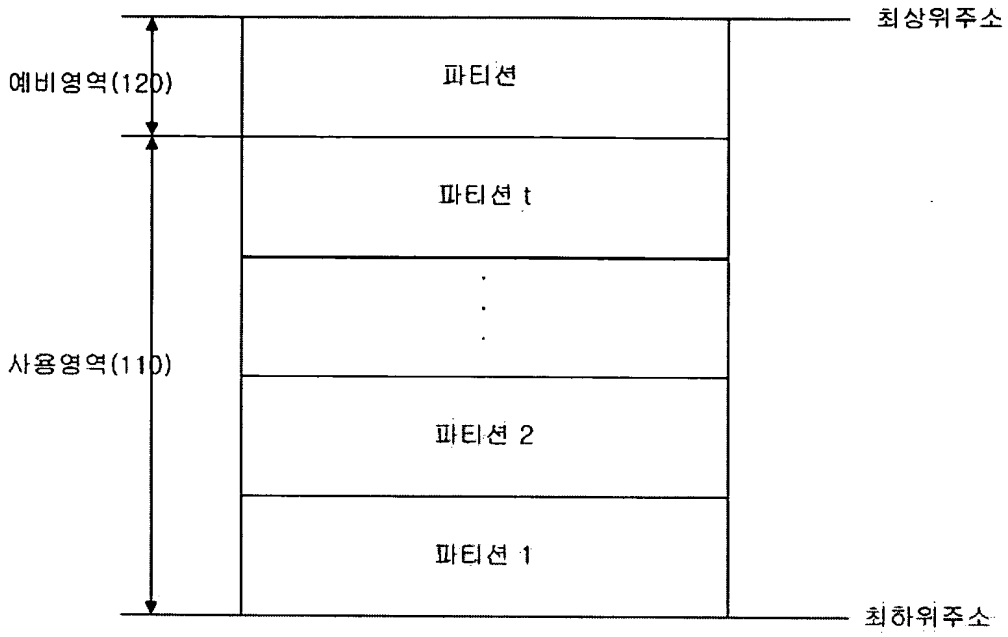
【도 3】


300
200
중앙처리장치
플래시 메모리 콘트롤러
400
100
오류블록관리 프로그램
플래시 어플리케이션
플래시 장치관리자
플래시 메모리

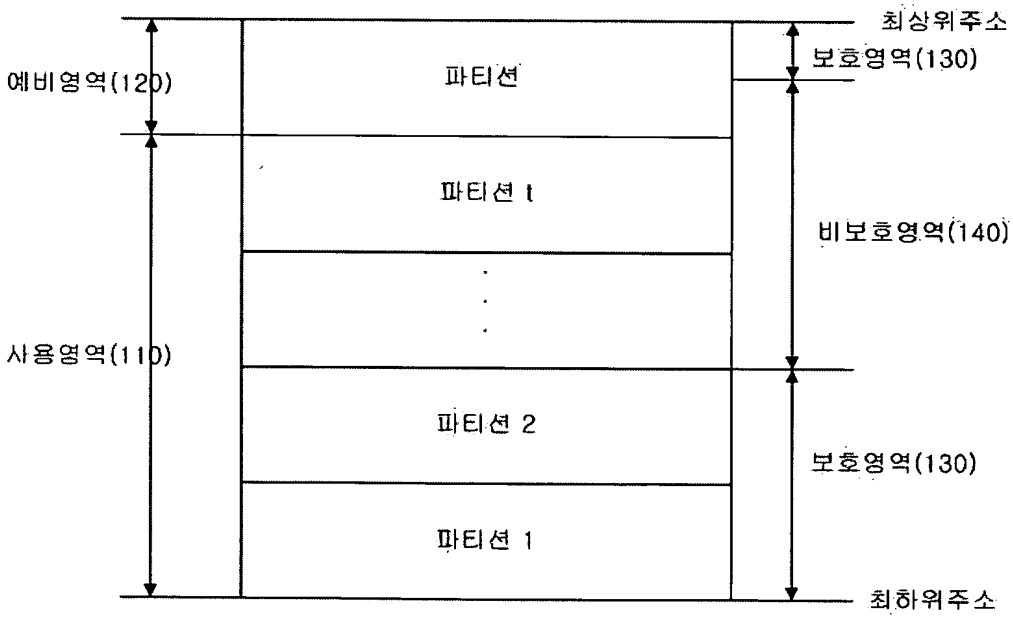
【도 4】



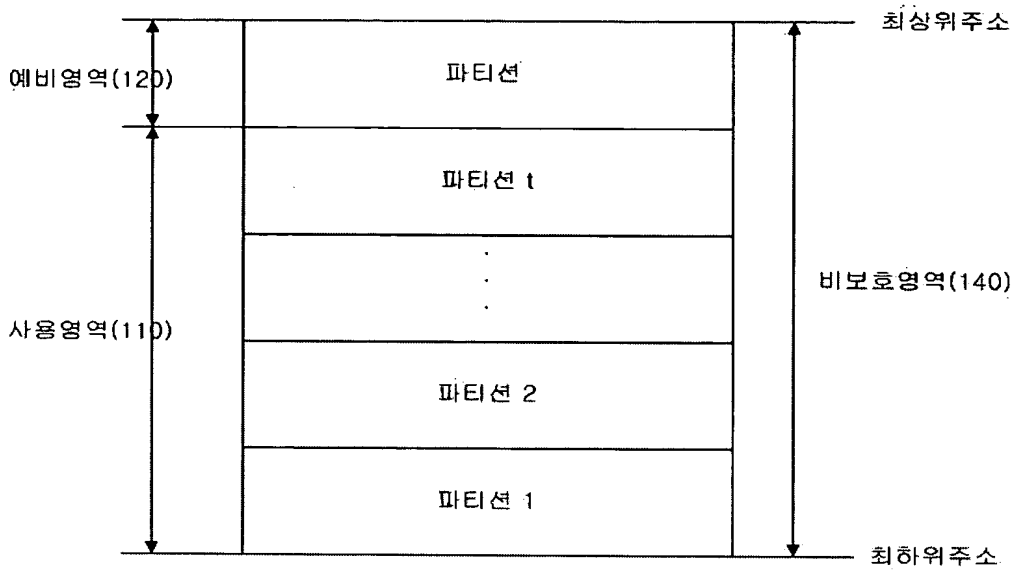
【도 5】


최상위주소
예비영역(120)
파티션
파티션 t
사용영역(110)
파티션 2
파티션 1
최하위주소

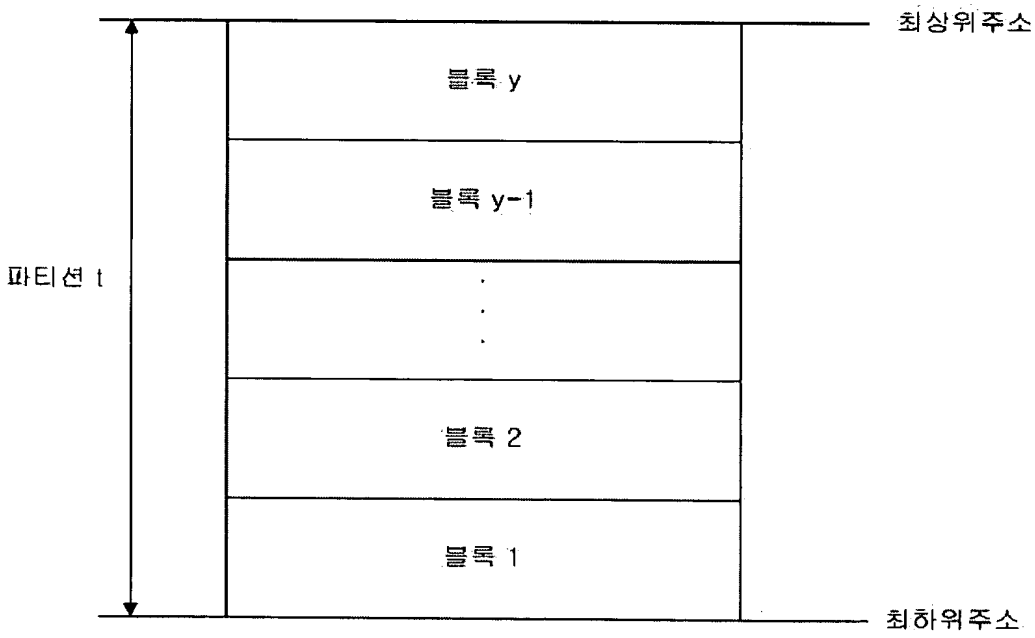
【도 6】


최상위주소
보호영역(130)
예비영역(120)
파티션
파티션 t
비보호영역(140)
사용영역(110)
파티션 2
보호영역(130)
파티션 1
최하위주소

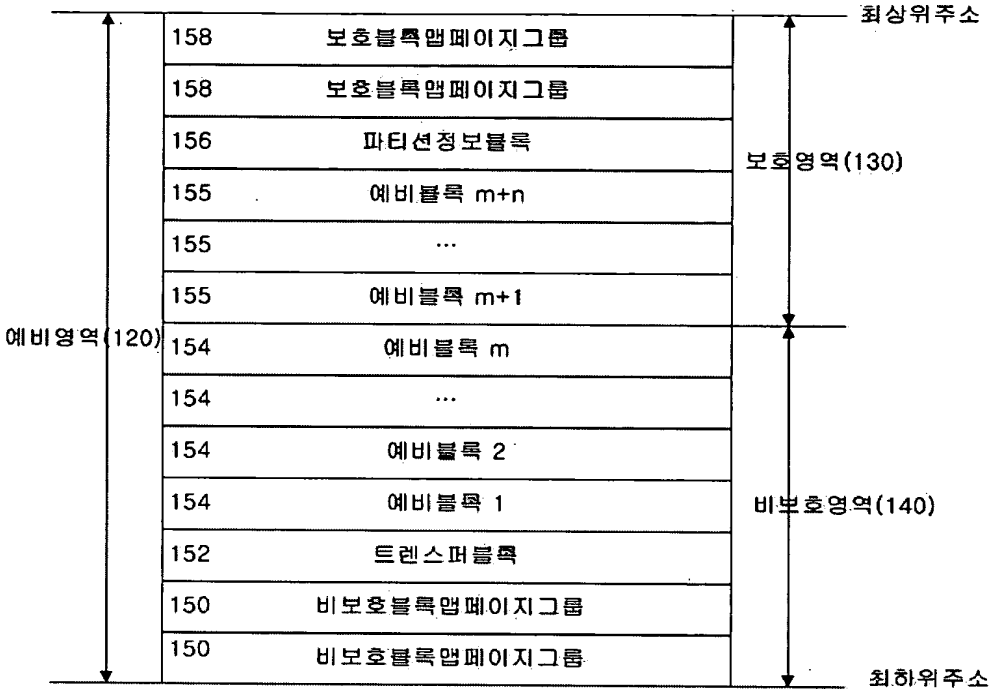
【도 7】


최상위주소
예비영역(120)
파티션
파티션 t
비보호영역(140)
사용영역(110)
파티션 2
파티션 1
최하위주소

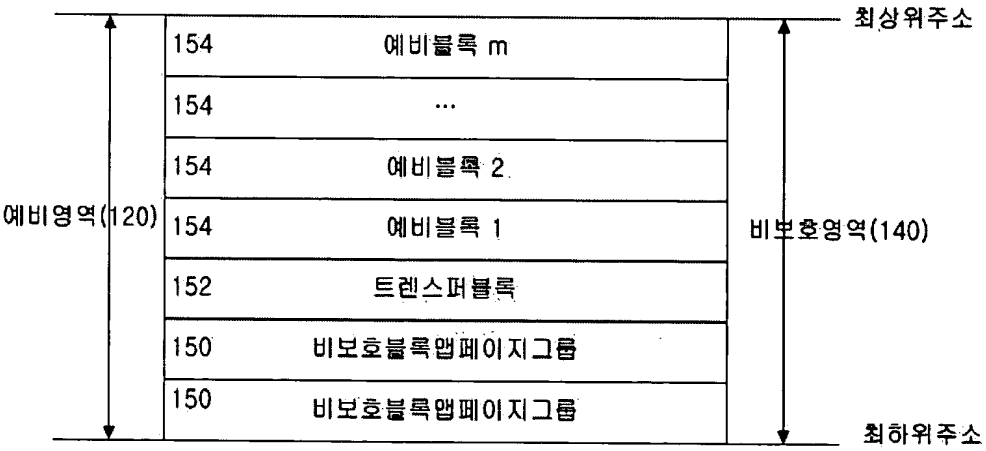
【도 8】


최상위주소
블록 y
블록 y-1
파티션 t
블록 2
블록 1
최하위주소

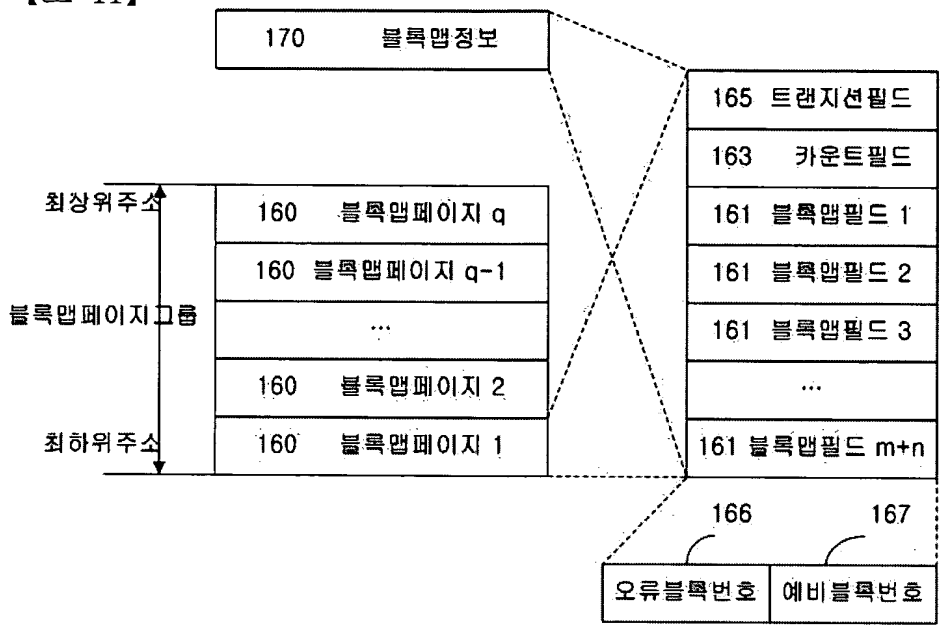
【도 9】


최상위주소
158 보호블록맵페이지그룹
158 보호블록맵페이지그룹
156 파티션정보블록
155 예비블록 m+n
155 ···
155 예비블록 m+1
보호영역(130)
예비영역(120)
154 예비블록 m
154 ···
154 예비블록 2
154 예비블록 1
152 트렌스퍼블록
150 비보호블록맵페이지그룹
150 비보호블록맵페이지그룹
비보호영역(140)
최하위주소

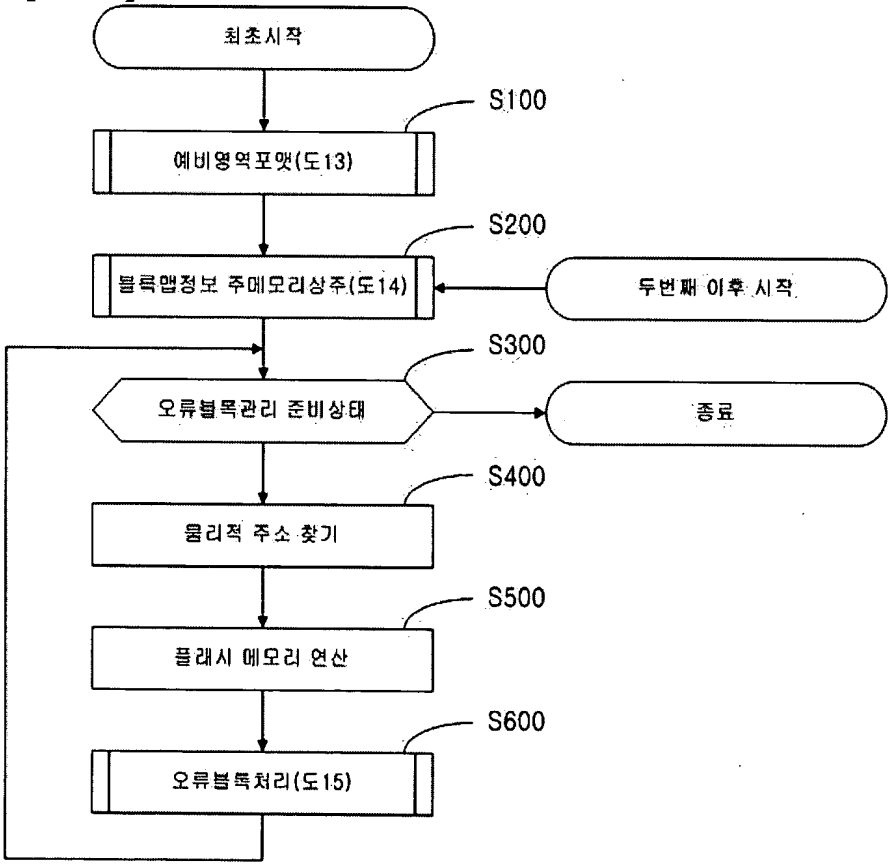
【도 10】


최상위주소
154 예비블록 m
154 ···
154 예비블록 2
154 예비블록 1
152 트렌스퍼블록
150 비보호블록맵페이지그룹
150 비보호블록맵페이지그룹
예비영역(120)
비보호영역(140)
최하위주소

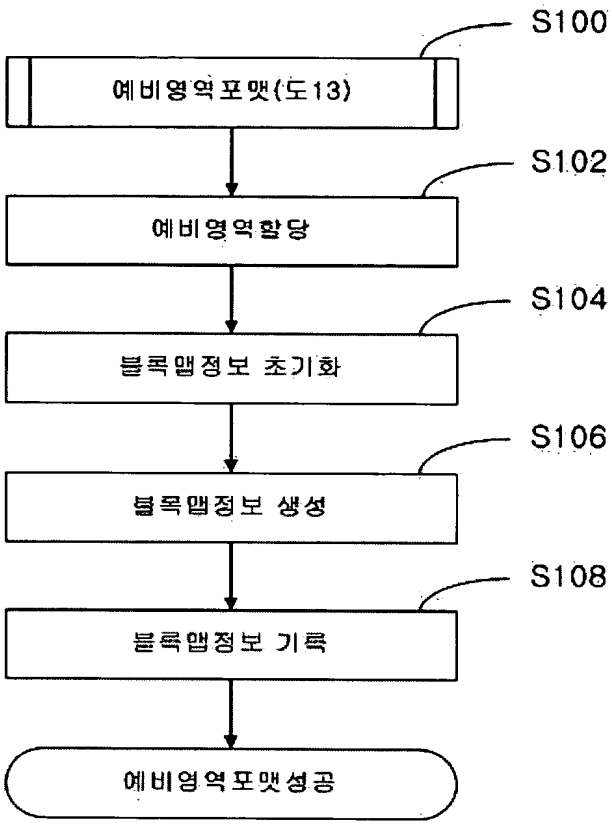
【도 11】


170 블록맵정보
165 트랜지션필드
163 카운트필드
161 블록맵필드 1
161 블록맵필드 2
161 블록맵필드 3
161 블록맵필드 m+n
166
167
오류블록번호
예비블록번호
최상위주소
블록맵페이지그룹
최하위주소
160 블록맵페이지 q
160 블록맵페이지 q-1
160 블록맵페이지 2
160 블록맵페이지 1

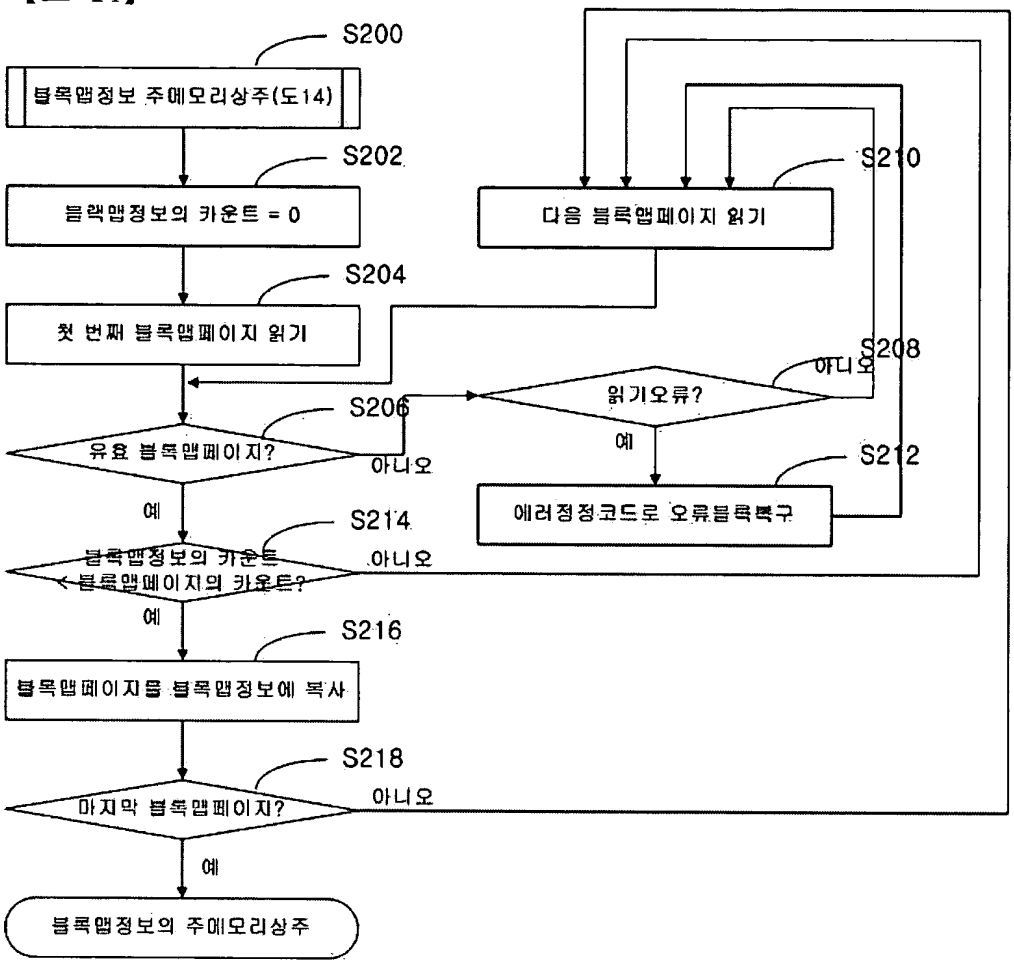
【도 12】


최초시작
S100
예비영역포맷(도13)
S200
블록맵정보 주메모리상주(도14)
두번째 이후 시작
S300
오류블록관리 준비상태
종료
S400
물리적 주소 찾기
S500
플래시 메모리 연산
S600
오류블록처리(도15)

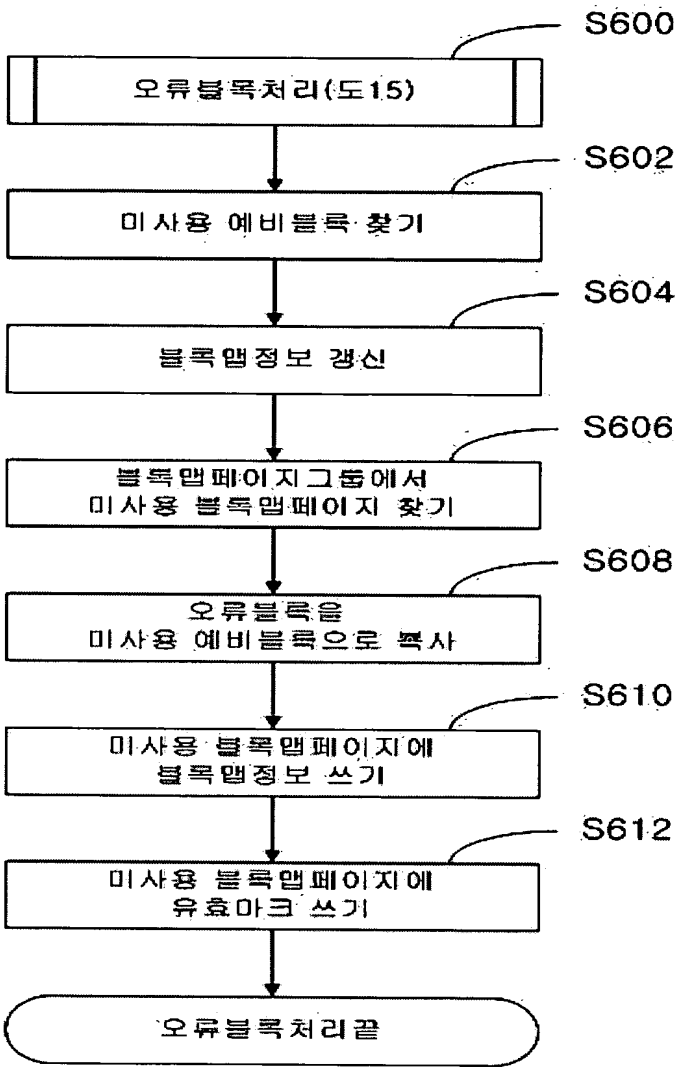
【도 13】


S100
예비영역포맷(도13)
S102
예비영역할당
S104
블록맵정보 초기화
S106
블록맵정보 생성
S108
블록맵정보 기록
예비영역포맷성공

【도 14】


S200
블록맵정보 주메모리상주(도14)
S202
블럭맵정보의 카운트 = 0
S204
첫 번째 블록맵페이지 읽기
S206
유효 블록맵페이지?
아니오
예
S208
읽기오류?
아니오
예
S210
다음 블록맵페이지 읽기
S212
에러정정코드로 오류블록복구
S214
블록맵정보의 카운트 < 블록맵페이지의 카운트?
아니오
예
S216
블록맵페이지를 블록맵정보에 복사
S218
마지막 블록맵페이지?
아니오
예
블록맵정보의 주메모리상주

【도 15】


S600
오류블록처리(도15)
S602
미사용 예비블록 찾기
S604
블록맵정보 갱신
S606
블록맵페이지그룹에서
미사용 블록맵페이지 찾기
S608
오류블록을
미사용 예비블록으로 복사
S610
미사용 블록맵페이지에
블록맵정보 쓰기
S612
미사용 블록맵페이지에
유효마크 쓰기
오류블록처리끝